0[18] Let \( n \) be a natural variable. You are given the refinement
\[
P \iff \text{if } n=0 \text{ then } n'>0. \text{ else } \text{ok}\fi
\]
Using recursive construction, find a solution for \( P \). You may ignore time.

1[12] Here is data-queue theory.
\[
(0) \quad \text{emptyq: queue}
(1) \quad \text{join q x: queue}
(2) \quad \text{join q x + emptyq}
(3) \quad \text{join q x = join r y = q=r \land x=y}
(4) \quad \text{emptyq, join B X: B ⇒ queue: B}
(5) \quad \text{leave (join emptyq x) = emptyq}
(6) \quad q + emptyq ⇒ leave (join q x) = join (leave q) x
(7) \quad \text{front (join emptyq x) = x}
(8) \quad q + emptyq ⇒ \text{front (join q x) = front q}
\]
Prove that if you start with an empty queue, and join two items, the first item joined is the front of the queue.

2 Implementer's variables \( p, q: \text{real} \) represent two points along a line. Each number tells the distance of one point from the origin (a standard point). They must be reimplemented by one implementer's variable \( r: \text{real} \) which tells the distance from \( p \) to \( q \). For examples, if \( p=3 \) and \( q=5 \), then \( r=2 \); if \( p=5 \) and \( q=3 \), then \( r=-2 \).

(a)[3] What is the data transformer? (no proof)
(b)[9] A user has binary variable \( b \) and operation \( \text{compare = b:= q \geq p} \)
Use your transformer from part (a) to transform operation \( \text{compare} \).